IN THE CLAIMS

1. A single integrated circuit (IC), comprising:

a synthesizable microcontroller processor core connected to a program code memory and able to selectively process one of two different program execution streams, and wherein a program-execution interrupt request forces a hardware switch to a first of said two different program execution streams, and further wherein the execution of a particular instruction is required to switch back to a second of said two different program execution streams;

an interrupt controller for receiving peripheral interrupt requests from a variety of system interrupt sources, and providing for said interrupt request to the core from a programmable combination of said peripheral interrupt requests; and

an interrupt service routine preamble;

wherein, said second of said two different program execution streams is more economical with program code space, so the interrupt service routine preamble is coded in said second of said two different program execution streams to cause a switch, and is shared amongst a plurality of interrupt service routines.

2. The IC of claim 1, wherein:

the interrupt controller provides for a fast interrupt request (FIQ) and a normal interrupt request (IRQ) to be programmably masked and prioritized before being issued to the core, and wherein said FIQ directs program execution to a last entry in an interrupt vector table.

3. The IC of claim 1, wherein:

the interrupt controller centralizes all interrupt handling and includes programmable interrupt masks for independent enabling and disabling each interrupt source, and for globally disabling all interrupts.

4. The IC of claim 1, wherein:

the interrupt controller further includes an interrupt vector control for automatically decoding a highest-priority interrupt for presentation of a programmed interrupt vector to the processor core.

5. The IC of claim 1, wherein:

the interrupt controller includes a global-disable control bit for use when a critical portion of program code is executing that is independent of any individual interrupt masks and avoids needing to save and restore mask states that would otherwise increase interrupt latency when a global-disable is lifted.

6. The IC of claim 1, further comprising:

a common part of an interrupt service routine (ISR) for putting the processor core an alternative program stream execution mode.

7. A single integrated circuit (IC), comprising:

a synthesizable ARM-type microcontroller processor core connected to a program code memory and able to selectively process either a THUMB or an ARM program execution stream, and wherein a program-execution interrupt request forces a hardware switch to said ARM program execution stream, and further wherein the execution of a BX-instruction is required to switch back to said THUMB program execution streams;

an interrupt controller for receiving peripheral interrupt requests from a variety of system interrupt sources, and providing for said interrupt request to the core from a programmable combination of said peripheral interrupt requests; and

an interrupt service routine preamble;

wherein, said THUMB program execution stream is more economical with program code space, and the interrupt service routine preamble is coded in ARM program code to

cause a switch to THUMB program execution, and the interrupt service routine preamble is shared amongst a plurality of interrupt service routines to further economize on program code space.

8. A method of processing a plurality of program-execution interrupt requests from system interrupt sources in a hardware environment that has program code space constraints, and in which an interrupt automatically is switched by processor hardware to an alternative program execution stream that is less efficient with program code space, the method comprising the steps of:

constructing an interrupt service routine preamble that includes a program instruction for returning to another program execution stream that is more efficient with program code space;

forcing an execution of said interrupt service routine preamble at a start of more-than-one interrupt service routines; and

executing within said interrupt service routine code belonging to said another program execution stream that is more efficient with program code space on a return to each said more-than-one interrupt service routines.